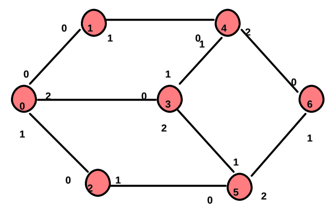
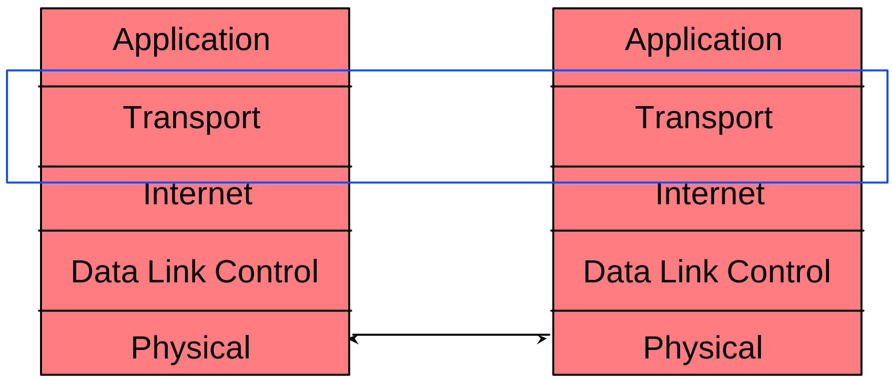
Transport Layer Protocols



Lecture Overview

- Transport layer functions
- UDP
- TCP
- SRMP

Internet Protocol Suite Reference Model

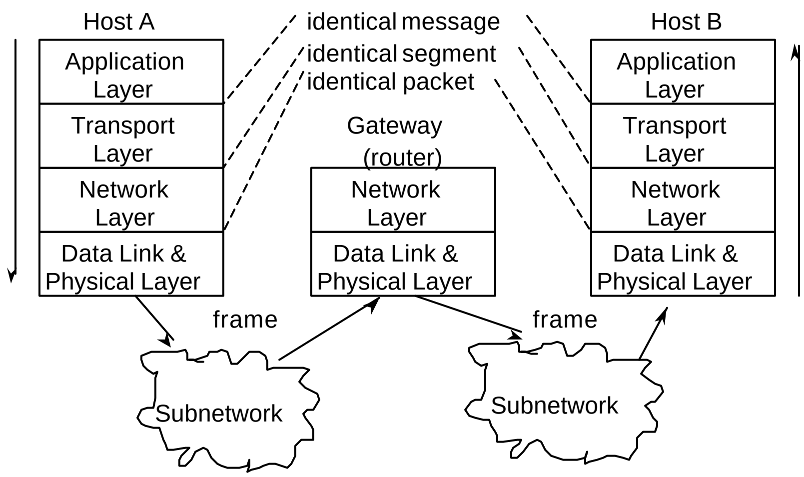


The DLC and Physical layers sometimes are referred to collectively as the “host to network” layer

Transport Layer Characteristics

- Transport Layer protocols reside in the source and destination nodes.
- Transport Layer provides a consistent service interface to the network.
- Transport Layer must provide reliable connection despite
  - Connectionless Network services
  - Virtual Circuit Resets on Connection-oriented Network Services
  - packet reordering that can occur when a Transport Connection is split across multiple Virtual Circuits

Transport Protocol
End-to-end Operation



Transport Layer *vs* DLC Layer

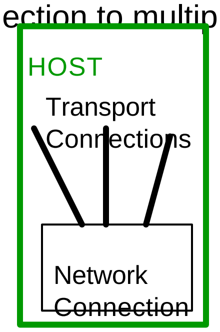
- Transport Layer and Data Link Layer bear some resemblance:
  - both perform error control, resequencing, and flow control.
  - Data Link layer operates on a link basis
  - Transport Layer operates end-to-end across a network

Elements Of Transport Protocols

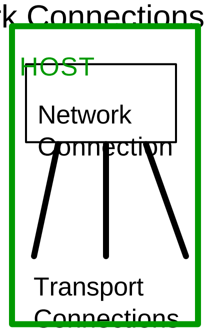
- Connection Management - establishment, refusal, and release. Manage mapping between transport and network connections or build a connection-oriented service from a connectionless network service. Link applications across a network.
- Segmentation and reassembly of application data - fragment data stream into packet sizes suitable for the network. Where multiple transport connections exist packets must be correctly associated with each connection. Receiver must reassemble packets and resequence the data stream to mirror what was sent.
- Recovery from network failures - reassignment after network disconnects and resynchronization after network resets.
- Error Control and Flow Control - especially for LANs.

Elements Of Transport Protocols
(continued)

- Multiplexing - mapping multiple Transport Connections to one Network Connection
- Splitting and Recombining - mapping one Transport Connection to multiple Network Connections.



Multiplexing



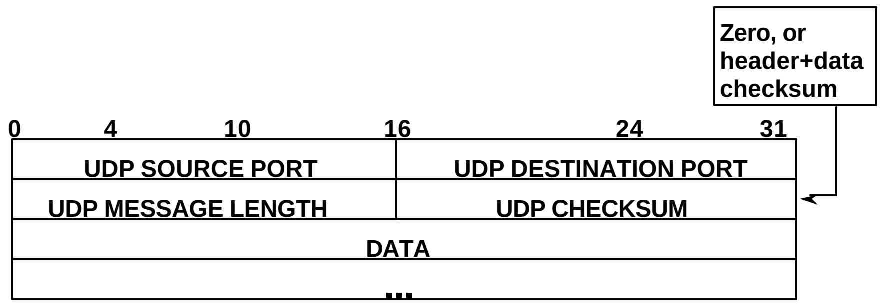
Splitting &
Recombining

Internet Transport Protocols

- UDP - User Datagram Protocol
  - ("unreliable data protocol") used for best-effort services
- TCP - Transmission Control Protocol
  - reliable transport

UDP

UDP Message Format



UDP Well-Known Ports

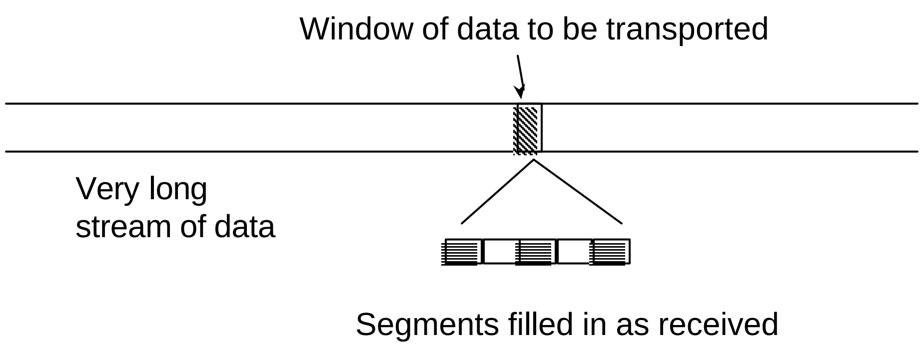
Decimal	*Keyword*	*Unix Keyword*	*Description*
0	ECHO	echo	Echo
7	DISCARD	discard	Discard
11	USERS	systat	Active Users
13	DAYTIME	daytime	Daytime
15	-	netstat	Who is up or NETSTAT
17	QUOTE	qotd	Quote of the Day
19	CHARGEN	chargen	Character generator
37	TIME	time	Time
42	NAMESERVER	nameserver	Domain Name Server
43	NICNAME	whois	Who is
53	DOMAIN	nameserver	Domain Name Server
67	BOOTPS	bootps	Bootstrap Protocol Server
68	BOOTPC	bootpc	Bootstrap Protocol Client
69	TFTP	tftp	Trivial File Transfer

TCP

Transmission Control Protocol

- TCP provides connection for connectionless IP
- reliable data transfer (selective repeat or go-back N)
- stream model of data
- maintains order in delivered stream
- flow control via sliding window
- multiplexing multiple sessions via "ports"
  - (Berkeley implementation: socket = IP address + port number)
- full-duplex transmission (ACK piggyback on reverse stream)
- one TCP connection is identified completely by:
  - (sending IPaddress/port,receiving IPaddress/port)
- precedence and security options - not widely used
- disconnect

TCP Sliding Window



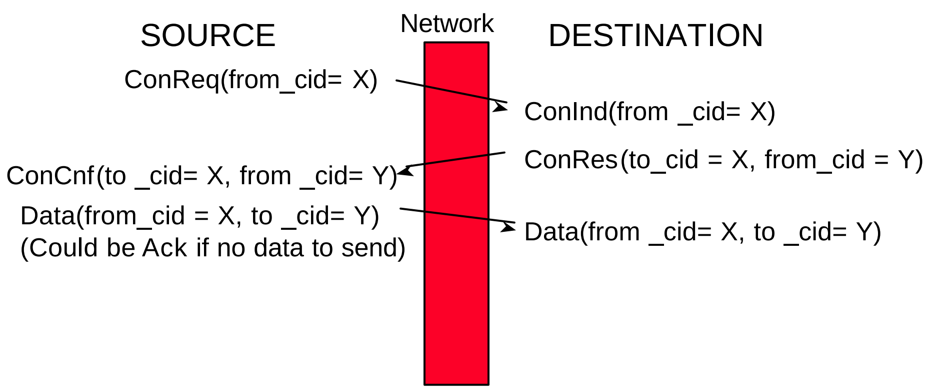
Connection Setup
Problem: Delayed Duplicate Sequence Numbers

- Packets may be delayed in the network and arrive out of order at the destination.
- This presents the possibility that an old packet from an earlier transport connection might arrive and appear to be in sequence for some current transport connection.
- Two mechanisms are commonly employed to guard against this:
  - Discarding packets within the network after a certain time (or number of hops).
  - Prohibiting sequence number reuse for a certain time. (In this case, the sequence number is a combination of a connection identifier and a packet identifier within the connection; thus, really the connection identifiers must be recycled in a time-sensitive manner.)

Transport Layer Connection Establishment

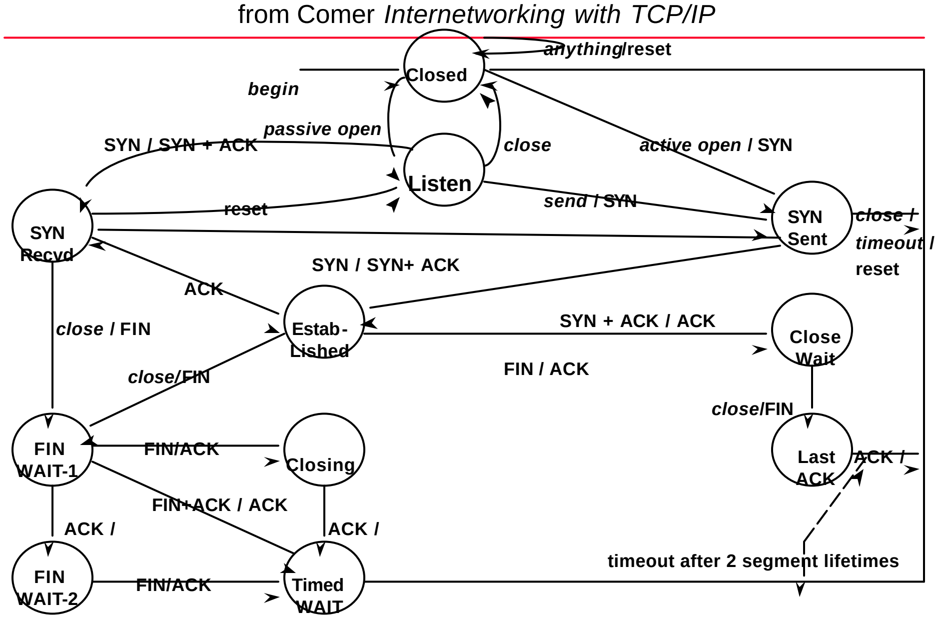
Three-Way Handshake

- The three-way handshake is used to avoid problems that can arise from delayed duplicate packets in connection setup.



TCP Connection State Machine

from Comer *Internetworking with TCP/IP*



TCP Segment Format

<table>
<tr><th>0</th><th>4</th><th>10</th><th>16</th><th>24 31</th></tr>
<tr><td colspan="3">SOURCE PORT</td><td colspan="2">DESTINATION PORT</td></tr>
<tr><td colspan="5">SEQUENCE NUMBER</td></tr>
<tr><td colspan="5">ACKNOWLEDGMENT NUMBER</td></tr>
<tr><td>HLEN</td><td>RESERVED</td><td>CODE BITS</td><td colspan="2">WINDOW</td></tr>
<tr><td colspan="3">CHEKSUM</td><td colspan="2">URGENT POINTER</td></tr>
<tr><td colspan="4">OPTIONS (IF ANY)</td><td>PADDING</td></tr>
<tr><td colspan="5">DATA</td></tr>
<tr><td colspan="5">...</td></tr>
</table>

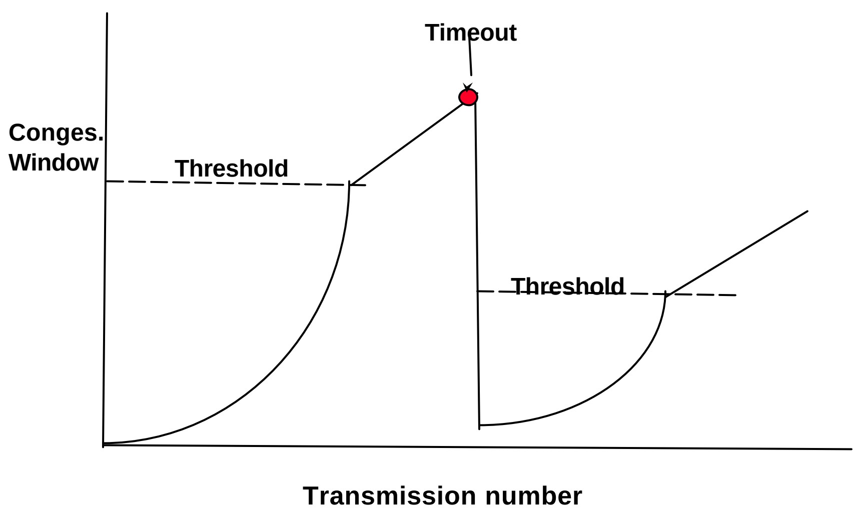
TCP Well-Known Ports

Decimal	*Keyword*	*Unix Keyword*	*Description*
0			Reserved
5	RJE	-	Remote job entry
11	USERS	systat	Active Users
13	DAYTIME	daytime	Daytime
15	-	netstat	Network status program
17	QUOTE	qotd	Quote of the Day
20	FTP-DATA	ftp-data	File Transfer Protocol (data)
21	FTP	ftp	File Transfer Protocol
23	TELNET	telnet	network terminal emulator
25	SMTP	smtp	Simple Mail Transfer Protocol
37	TIME	time	Time
42	NAMESERVER	nameserver	Host Name Server
43	NICNAME	whois	Who Is
53	DOMAIN	nameserver	Domain Name Server
80	WWW-HTTP	http	World Wide Web HTTP

TCP Operation

- Code bits: used to originate connection
- Window: size varied for congestion control (reduce multiplicatively, increase additively)
- Round-trip time (RTT): TCP keeps track of time from SEND to ACK
- Timeout: if some multiple of RTT passes with no ACK, send again
- Establishing connection: requires 3-way handshake (one end must perform "passive open")
- Well-known ports: reserved for functions (e.g. email) or "meeting points"
- Push: issued by sending application to cause TCP to send a segment (important for interactive applications)

TCP Congestion Control Example



TCP Startup

- Port selection:
  - application has pre-established “meeting” port
  - if multiple sessions are possible for the application, can use initial connection to agree on a new “working” port
- Slow-start:
  - to avoid congestion
  - start with minimum window
  - double window size each round-trip
  - if no congestion seen (by increased RTT), continue doubling until configuration max reached

TCP Retransmission Timer

- Used to decide when to retransmit segment.
- Problem: difficult to determine correct value for time-out over WAN.
  - if the timer is too short, unnecessary retransmissions occur, clogging up the network.
  - If the timer is too long, performance suffers due to long retransmission delays.
- Solution: use a highly dynamic algorithm that constantly adjusts the timeout interval.

TCP Timeout Algorithm
continued

- Problem: when an acknowledgement comes in, it is unclear whether the acknowledgement refers to the first transmission or the later one.
- Solution (Karn's algorithm): do not update RTT on any segments that have been retransmitted. Instead, double the timeout.

TCP for Interactive Applications

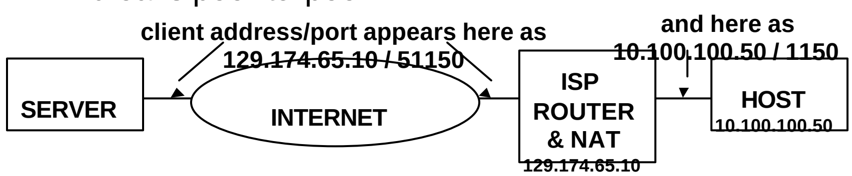
- Interactive applications such as telnet may need to send as little as one byte of data in a segment
- "push" feature allows the application to tell TCP to send data without waiting to accumulate a full segment
- When network response is slow, sending one byte per segment would be inefficient; use Nagle's algorithm to avoid this:
  - while waiting for an ACK, accumulate send data in buffer
  - when ACK arrives, send contents of the buffer

Limitation of TCP for High- Performance Networks

- TCP's slow-start and "knee-jerk" congestion control can cause a problem in the emerging class of gigabit/second-backbone networks
  - for example, Abilene and DREN
- The problem occurs where distance makes latency high
  - only a problem when "delay-bandwidth product" is large
- If a single packet is lost, TCP performance drops 50% and grows back very slowly
- IETF is considering options that may work better in high-performance networks

Network Address Translation

- As presented previously, IPV4 is running short of addresses
  - but ISPs are reluctant to make the major change to IPv6
- A common work-around Network Address Translation (NAT)
  - use one IPv4 address to "front" for many hosts
  - assign hosts addresses from "private" address space
  - for example, class A net 10 (ARPANET) which is set aside for testing
  - run a process that uses TCP port multiplexing to identify the "back side" host address
  - substitutes the NAT's IP address in the packet header
  - only works when the host initiates connection
    - breaks peer-to-peer



SRMP

GMU SRMP

How to achieve Reliable Multicast?

- No “one size fits all” RM protocol like TCP
  - solutions tailored to application domain
  - essence of the problem is functionality vs congestion
- Methods poorly understood and not generic:
  - must meet needs of application domain (for example, distributing a file to many locations - multicast FTP)
  - limiting network congestion to sustain performance (for example, NACK implosion when many stations detect the same lost packet)

SRMP Concept

Reliable Multicast for NVE

Networked virtual environments have two important characteristics that RM can use to advantage:

- Only the latest value of any object attribute variable is required by the simulation
- Some object attributes change rarely (e.g. appearance) while typically a few attributes change frequently and so are transmitted at regular intervals (e.g. position)

Based on these characteristics we have concluded:

- A single transport protocol can achieve synergy for NVE by combining RM of rarely-changing data with best-effort transmission of frequently-changing data

Performance Testing Via WAN Emulation

- Emulate target system on Myrinet network of high-performance workstations
- Latency represented by configurable delay
- Statistical packet discard

Federate(s)
RTI
SRMP
WAN Emulation
UDP/IPmc

SRMP Performance

Variable	Value
Number of WAN locations	4
Number of objects per location	15
Duration	30 minutes
Packet size	2000 bits
Simulated packet drop	5%
Portion of time objects active	50%
Mean period of activity	50 seconds
Idle state frequency	0.5 packets/sec
Moving state frequency	5 packets/sec
Mode 0 packets transmitted	296,684
Mode 1 packets transmitted	2,169
Mode 1 retransmissions	2,220*

*due to 6,500 lost

Reliable Multicast and Congestion

- Protocols for shared network use need to avoid causing congestion
- The Internet standard TCP does this by reducing transmit rate when round-trip time increases
  - RM protocols can't use this mechanism
  - but the IETF requires them to be "TCP friendly"
  - proposed approach TCP Friendly Multicast Congestion Control (TFMCC) Internet-Draft by Widmer & Handley
- Approach planned for SRMP:
  - NACK suppression and message bundling
  - sense congestion with TFMCC, drop some best-effort traffic
  - tree-based logging and repairs